I. J. Computer Network and Information Security, **2017, 6, 53-60**
Published Online June 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2017.06.06

A Hybrid Cryptographic Model for Data Storage in Mobile Cloud Computing

Oladeji P. Akomolafe
University of Ibadan/Department of Computer Science, Ibadan, Nigeria
E-mail: akomspatrick@yahoo.com

Matthew O. Abodunrin
University of Ibadan/Department of Computer Science, Ibadan, Nigeria
E-mail: mabodunrin8094@stu.ui.edu.ng

Abstract—Mobile Cloud Computing (MCC) is a paradigm that integrates Cloud Computing and Mobile Computing to deliver a better Quality of Experience (QoE) and Quality of Service (QoS) to mobile users and cloud subscribers. Mobile Cloud Computing (MCC) inherited resource limitation from Mobile Computing which was solved with Cloud Computing. Meanwhile, Cloud Computing has inherent problems such as privacy of user's data stored on cloud, intrusion detection, platform reliability, and security threats caused by multiple virtual machines. Thus, hindering the growth and the full acceptance of Mobile Cloud Computing (MCC) by subscribers. However, using a signature based hybrid cryptography ensures confidentiality, integrity, authentication and non-repudiation on resource-poverty devices used in Mobile Cloud Computing. This paper presents a data protection scheme where data is encrypted using a hybrid cryptographic algorithm which is composed of Advanced Encryption Standard (AES), Blake2b and Schnorr signature before being stored in the cloud storage (Amazon Simple Storage Server). Thus, data confidentiality, integrity, authentication and non-repudiation are ensured.

Index Terms—Mobile Cloud Computing (MCC), Amazon S3, Advanced Encryption Standard (AES), Blake2b, Schnorr signature, Cryptography.

I. Introduction

The ubiquity and increasing capacity of mobile devices have ignited many new computing paradigms such as Mobile Computing, Mobile Cloud Computing, Mobile Sensing, Pervasive Computing, Collaborative Computing and so on. Mobile Cloud Computing (MCC) as one of the paradigms which amalgamated Cloud Computing with Mobile Computing was defined by [2] as "the combination of cloud computing, mobile computing and wireless networks to bring rich computational resources to network operators, mobile users and cloud computing providers".

MCC inherited different problems from the comprising paradigms. It inherited resource-poverty from Mobile Computing which was solved with Cloud computing. Meanwhile, Cloud Computing has inherent problems such as security and privacy of user's data that is stored on cloud, intrusion detection, platform reliability, and security threats caused by multiple virtual machines [12].

In order to mitigate the security and privacy challenges, cryptography has been employed over the years to provide privacy, confidentiality and integrity. Cryptography as one of the major approaches to solving data security issues is defined as the process of encrypting and decrypting data to ensure privacy, confidentiality and integrity. The major feature goals of cryptography are: confidentiality, data integrity, authentication and non-repudiation. These are achieved by using the following cryptography schemes: Encryption, Hash Functions, Message Authentication Codes (MAC) and Digital Signatures. There are three types of Cryptography: Symmetry, Asymmetry and Hybrid [22].

This paper presents a hybrid cryptographic model called ABS comprising Advanced Encryption Standard (AES) Blake2b and Schnorr Signature Scheme which is resource-efficient for better data security in Mobile Cloud Computing (MCC).

A. *Mobile Cloud Computing*

MCC is an integration of Mobile Computing and Cloud Computing. Mobile Cloud Computing integrates the basic functional architectures of mobile communication using either wireless access point or radio tower (Wi-Fi, 3G, 4G or 5G technologies) to access cloud resources (Servers and Virtual Machines) [7] due to resource limitations of the mobile devices as shown in the Fig. 1 below

B. *MCC Model*

MCC model consists of three components mainly as shown in Fig.1. These components include [23]:

- *Mobile Terminal*: This refers to the portable and movable devices such as smart phones, PDA and tablets.
- *Cloud:* It contains servers providing services like Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS).

- *Mobile Network:* It makes services available to mobile devices. So whenever we talk about the security in MCC, we should consider the security and privacy issues in all these three aspects.

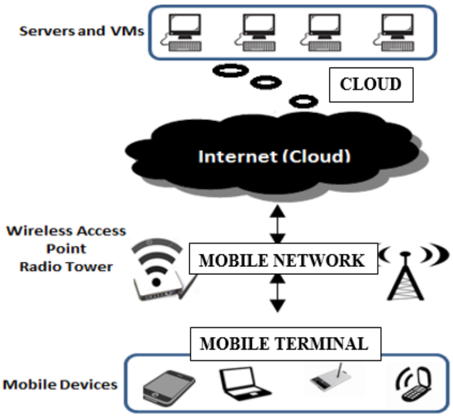

Fig.1. Overview of Mobile Cloud Computing [17]

C. *MCC Issues*

In Mobile Cloud Computing, the data storage and processing are done in the cloud [15]. The security and privacy of user's data that is stored on cloud is one of the major issues needed to be resolved in MCC [12] because MCC users may store all confidential data in the cloud for either a short or long term usage with little or no control over it while a malicious mind in disguise of a cloud staff may access the confidential data, steal and pass it on to user's competitors [19]. Therefore, the security of the data in the cloud becomes a real issue. Moreover, the user data needs to be protected not only from the outside attackers but also make it inaccessible to the cloud service providers.

D. *Cryptography*

Reference [6] defined "Cryptography as the science of using mathematics for making plain text information (P) into an unreadable cipher text (C) format called encryption and reconverting that cipher text back to plain text called decryption with the set of Cryptographic Algorithms (E) using encryption keys (k1 and k2) and the decryption algorithm (D) that reverses and produces the original plain text back from the cipher text. This can be interpreted as Cipher text C = E {P, Key} and Plain text C = D {C, Key}".

As depicted in Fig. 2 below, the user data and the key undergo series of enciphering through cryptographic algorithms to generate the cipher text called encryption. Likewise, the cipher text and the key are used to generate the original user data through the same cryptographic algorithms called decryption.

E. *Cryptographic Schemes*

These are the disciplines under cryptography that made the achievement of the goals of Cryptography (Confidentiality, Data Integrity, Authentication, Non-repudiation, and Access Control) [10] feasible. These include:

1) *Encryption and Decryption:* This is majorly conversion of plaintext to ciphertext and ciphertext back to the original plaintext. The goal of confidentiality or privacy is achieved with this scheme. Some of the algorithms are RC6, 3DES, Blowfish, AES, ElGamal, RSA, ECC and Diffie Hellman [21].
2) *Hash Functions:* These are transformations which produce a numeric value called "hash" or "digest" of fixed size from an input message of arbitrary length (both typically measured in bits). These are often used for checking the integrity of transmitted data, therefore, hash functions are means to achieving the goal of integrity. Examples include MD4, MD5, SHA 1, SHA 2, SHA 3 and Blake2b [21].
3) *Message Authentication Codes (MACs):* These are symmetry short codes called "MACs" or "tags" being used for authenticating a message. They are used to achieve integrity and authentication. They are different from digital signatures because they use a single key just like the symmetry algorithms and cannot provide non-repudiation. The algorithms are HMAC, DAA, and CMAC [21].
4) *Digital Signatures:* These are electronic signatures mimicking written signatures to provide authentication, access of data, non-repudiation and integrity. The algorithms for digitaal signatures include ElGamal Signature, Schnorr Signature, DSA, RSA and ECDSA [21].

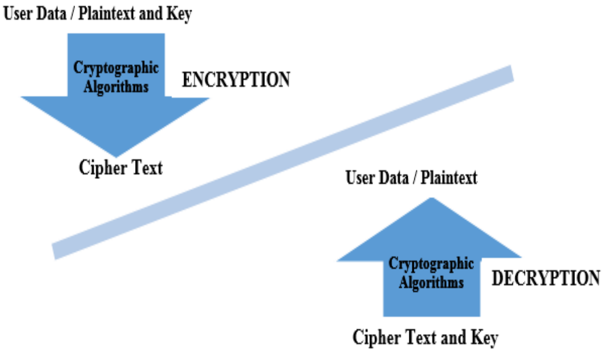

Fig.2. Overview of Cryptography

F. *Cryptographic Algorithms*

The following algorithms made up the hybrid cryptographic scheme:

1) *AES:* This is a symmetry algorithm based on Substitution-Permutation Network (SPN) which processes the entire data block as a single matrix during each round using substitutions and permutation algorithm by two cryptographers from Belgium, Joan Daemen and Vincent Rijmen. The plaintext undergoes four basic transformations. Each of the transformation takes one or more 4 by 4 matrices [21].

For encryption, it has four basic transformation steps [19]:

a) *Substitute Bytes (SubBytes):* Uses an S-box to perform a byte-by-byte substitution of the block in order to resist linear and differential cryptanalysis attack.
b) *ShiftRows:* A simple permutation where each row is shifted by one byte in a cyclical manner.
c) *MixColumns:*A substitution which involves matrix multiplication of one column at a time.
d) *AddRoundKey:* The matrix is XORed with the round key.

For decryption, an inverse function of each of the transformation is used with the expanded key in reverse order. It generates the ciphertext from the plaintext input providing confidentiality for the data [21].

2) *Blake2b Algorithm***:** The BLAKE function was designed by a team of four individuals based in Switzerland. Its principal designer was Jean-Philippe Aumasson [4]. It is an iterative algorithm based on the well-known and highly-regarded Hash Iterative Framework (HAIFA) structure [8]. It is used for creating checksum or hash value or digest in order to check for any alteration in the data. Blake2b is one of the fastest hash functions [5][18]
3) *Schnorr Signature Scheme:* This algorithm is used for non-repudiation, authentication and verification providing a better integrity. Its security is based on the intractability of certain discrete logarithm problems. It is considered the simplest digital signature scheme to be proveably secure in a random oracle model [20]. The hash value is signed instead of the whole plaintext in order to reduce the running time of the algorithm. The signature has the following steps: Key generation, Signing and Verifying.

II. Related Works

The researchers developed a scheme for data encryption to protect sensitive data leakage. A hybrid approach of Elliptic Curve Cryptography (ECC) and Blowfish Algorithm for mobile platform was proposed by integrating some modifications in Blowfish encryption/decryption to protect the data from being disclosed. This approach is deficient because the data integrity is not secured. It lacks non-repudiation, privacy and also Blowfish cannot be used for a large data or video [16].

A mechanism to provide the three security primitives- confidentiality, integrity, and authentication by combining the cryptographic encryption algorithms (RSA and AES) as the hybrid algorithm, and hash functions was proposed. However, RSA is very slow, the integrity of the data with SHA256 is not efficient without a digital signature. This hybrid lacks authentication because authentication and integrity can be achieved with digital signature while SHA256 is mainly used for creating hash value or digest of the plaintext [1].

The data security was improved by proposing an architecture that integrates the Cryptographic Algorithms, Advanced Encryption Standard (AES) algorithm and the Hash function, SHA-2 [24]. However, it lacked authentication, access of data, non-repudiation and integrity due to the absence of digital signature.

Reference [22] proposed an approach to securely store the mobile data in cloud using minimal performance degradation. They designed three-tier hybrid approach of MD5, AES and ECC algorithm. In this, encryption was done via MD5 algorithm, then this encrypted data was again encrypted by AES algorithm, and at last further encryption was done by ECC or RSA algorithm. This approach lacked digital signature and MD5 is best used as a hash function other than encryption because of its security vulnerability.

A new hybrid cryptographic algorithm using combination of two symmetric cryptographic techniques and two asymmetric cryptographic techniques was proposed by [9]. This provides three cryptographic goals- integrity, confidentiality and authentication. It is a hybrid encryption method where Elliptical Curve Cryptography (ECC) and Advanced Encryption Scheme (AES) are combined to provide node encryption. (RSA) Algorithm and (Blowfish) are combined to provide authentication and (MD5) for integrity. This approach is costly due to overhead cost of computation for the mobile devices. Even though, it would provide a hard-to-break security. The overhead cost would be high since RSA and Blowfish have high overhead cost.

A new security protocol was proposed using hybrid cryptographic algorithms of AES, ECC and XOR-DUAL RSA for encryption with MD5 hash function and AES, ECC and XNOR-DUAL RSA for decryption with MD5 hash function [3]. Nonetheless, the execution time of this algorithm is long because the plaintext is encrypted sequentially by both AES and ECC, even though, it has a parallel execution.

III. Methodology

The hybrid cryptographic model (ABS) includes AES (128 bits key), Blake2b and Schnorr Signature Scheme. Fig. 3 shows the proposed MCC storage architecture where mobile devices run the ABS algorithm and communicate with the cloud data storage via mobile network or WIFI.

Reference [26] suggested a personal data encryption unknown to the cloud service provider in order to ensure a better level of security, privacy and trust for the user. Hence, the encryption of data is done on the mobile device using ABS before uploading to Amazon S3 cloud storage environment and also decryption is done on the mobile device after downloaded from the cloud storage.

The model has two major architectures: encryption and decryption.

A. *Encryption Architecture*

Fig. 4 shows the architecture for the encryption. It involves:

- *Key Generation:* The public key and the private key are generated for Schnorr signature.
- *Secret Key Generation*: The user data or plaintext is hashed with Blake2b to generate data hash. The private key is XORRED with the data hash to generate the SECRET KEY for the AES.
- Sign the data hash using the generated private key with Schnorr Scheme (SIGNATURE GENERATED).
- Generate the round keys by expanding the SECRET KEY.
- Add the round keys with user data or plaintext.
- Perform the basic AES operations (Substitute bytes, Shift rows, Mix columns, Add random key).

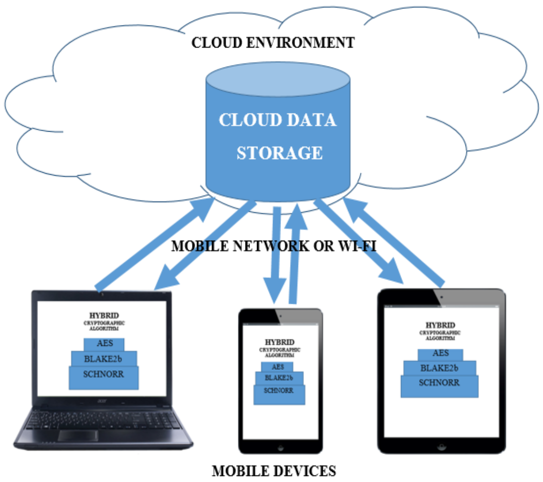

Fig.3. Mobile Cloud Computing Storage Architecture

- Repeat previous step for 8 times.
- Perform the basic AES operations excluding Mix columns only (CIPHER TEXT GENERATED).
- Upload the SIGNATURE GENERATED and CIPHER TEXT GENERATED to the cloud data storage.

B. *Decryption Architecture*

The ciphertext is downloaded from the cloud storage after a valid verification of the signature uploaded with the particular encrypted file while it terminates if the verification is invalid

The decryption is done on the mobile device after a successful verification and downloading of the encrypted file or cipher text. Fig. 5 shows the architecture for the decryption process of the cryptographic model. The steps include:

- Verifying the signature using Schnorr Verification Scheme with the hash value, public key and the signature generated while encrypting.
- Terminate if verification is unsuccessful or else go to the next step.
- Input the secret key generated during the encryption for AES.
- Generate the round keys by expanding the SECRET KEY.
- Add the round keys with ciphertext.
- Perform the basic AES inverse operations (Inverse Substitute Bytes, Inverse Shift Rows, Inverse Mix Columns, Add Random Key).
- Repeat previous step for 8 times.
- Perform the basic AES inverse operations excluding Inverse Mix columns only (USER DATA DECRYPTED).

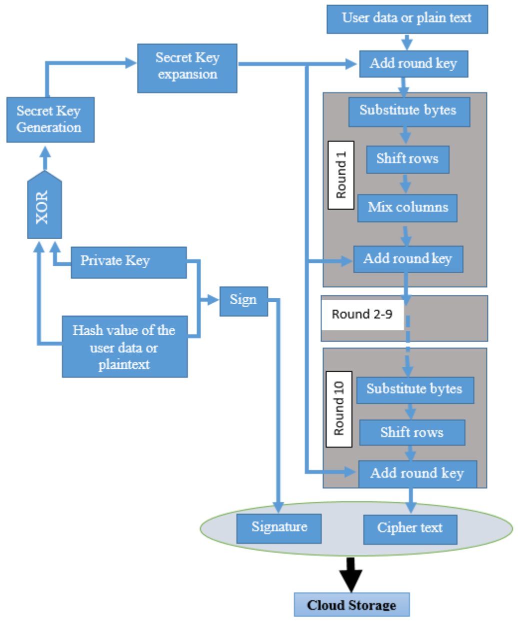

Fig.4. ABS Encryption Architecture

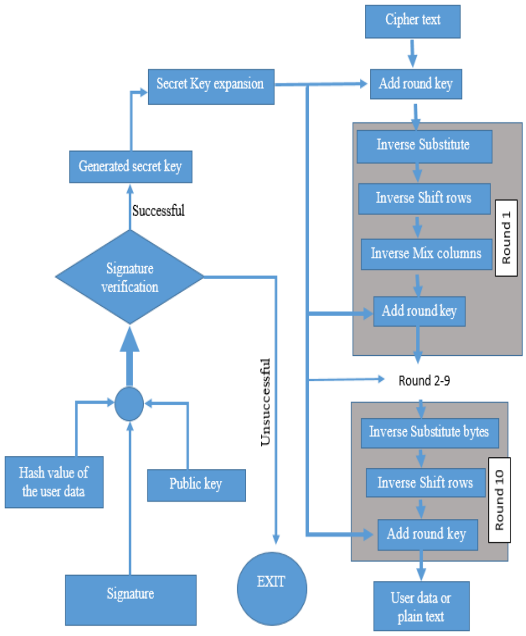

Fig.5. ABS Decryption Architecture

IV. Implementation Results

Amazon S3 was used for the cloud storage environment. The following algorithms- ABS, AES and Blowfish were implemented on an android device (Processor- 8x ARM Cortex-A7 @1.36GHz, Android version 4.4.2 (API 19) and RAM – 1GB) using different file sizes ranging from 1MB-200MB.

The results of ABS observed were compared with the selected symmetry algorithms - AES and Blowfish. The metrics used for the comparison are the encryption and decryption execution times, percentage computation overhead and the throughput as shown in tables I - III and illustrated in Fig. 6-11. Fig.5 compared the encryption and decryption execution time of ABS.

The comparison of the execution time for both encryption and decryption of sample data sizes for the selected algorithms was analyzed in table 1 below.

Table 1. Comparison of Execution time for Encryption and Decryption Time of the Algorithms on Smart Phone

File Size (MB)	Execution time (s)					
	AES		*Blowfish*		*ABS*	
	*EN**	*DE**	*EN**	*DE**	*EN**	*DE**
1	0.743	1.454	0.741	1.416	0.787	1.454
5	3.389	10.054	2.950	8.447	3.974	14.107
10	4.172	13.582	4.806	14.291	4.309	15.264
25	10.539	38.966	11.717	42.831	11.041	40.202
50	20.191	76.230	27.033	85.422	20.879	80.866
100	45.204	153.662	68.831	178.918	51.426	162.412
150	71.291	213.536	96.794	231.182	78.186	219.143
200	96.539	312.285	123.571	327.000	98.484	313.050

The developed ABS performed considerably very close to AES but faster than Blowfish for file size greater than 10MB in terms of encryption and decryption times as shown Fig. 6 and 7 respectively. Perhaps, if AES and ABS are applied to a larger file size, the time difference would be totally negligible.

Fig. 6 is the graphical representation of the execution time of ABS showing the encryption and decryption execution times. The time of execution for the encryption and decryption processes of ABS increases with the increasing file size. The encryption-decryption speed ratio ranges from 1.91 to 3.87 approximately.

Fig. 7 displays the encryption time comparison of ABS, AES and Blowfish algorithms. ABS performed considerably very close to AES but faster than Blowfish for file size greater than 10MB. Perhaps, if AES and ABS are applied to a larger file size, the time difference would be negligible.

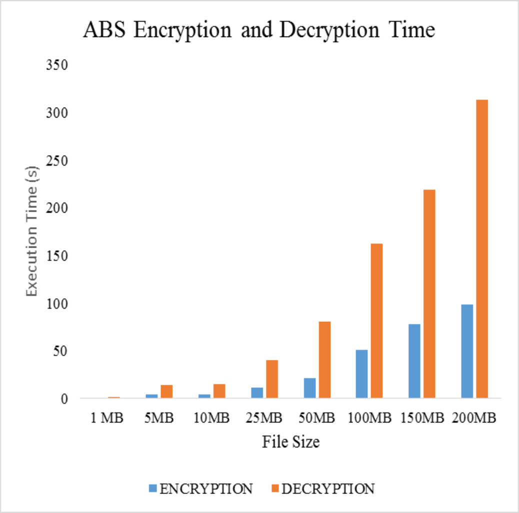

Fig.6. ABS Execution Time

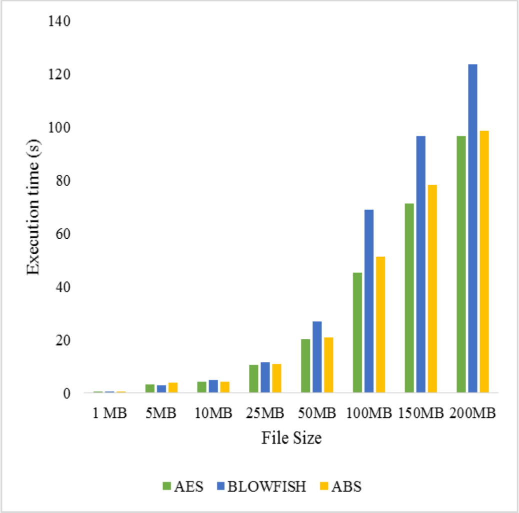

Fig.7. Encryption Time Comparison

The decryption time comparison is represented in Fig. 8 below.

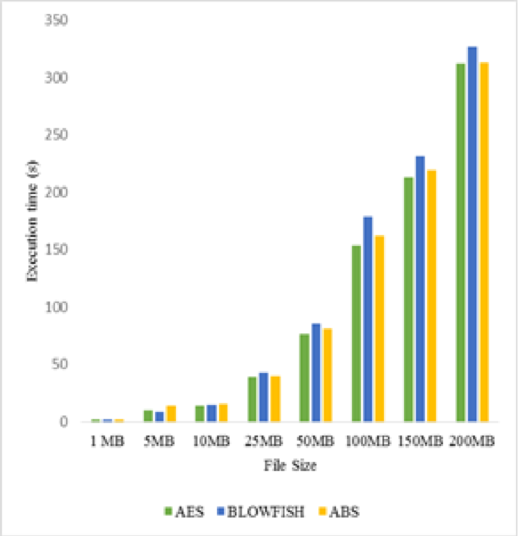

Fig.8. Decryption Time Comparison

EN*- Encryption, DE*- Decryption

The throughput (file size per unit execution time) of each of the algorithms for both encryption and decryption was computed and compared according to the table 2 below.

Table 2. Throughput Comparison for Encryption and Decryption of AES, ABS and Blowfish

Throughput (MB/s)						
File Size (MB)	***AES***		***Blowfish***		***ABS***	
	EN*	***DE****	***EN****	***DE****	***EN****	***DE****
1	1.345	0.688	1.345	0.706	1.271	0.674
5	1.475	0.497	1.695	0.592	1.258	0.354
10	2.397	0.736	2.081	0.670	2.321	0.655
25	2.372	0.641	2.134	0.584	2.264	0.622
50	2.476	0.656	1.850	0.585	2.395	0.618
100	2.212	0.651	1.452	0.559	1.945	0.616
150	2.104	0.702	1.550	0.648	1.919	0.684
200	2.071	0.640	1.619	0.611	2.031	0.639

The throughput of the algorithms was calculated using [25]:

$$Throughput = t_p / e_t \, , \qquad (1)$$

where t_p is the file size (MB) and e_t is the execution time.

As shown in Fig. 9 and 10 for encryption and decryption respectively, ABS has a better throughput than Blowfish at file size greater than 5MB during encryption and also at file size greater than 10MB during decryption while it has a close throughput with AES at large file size during both encryption and decryption.

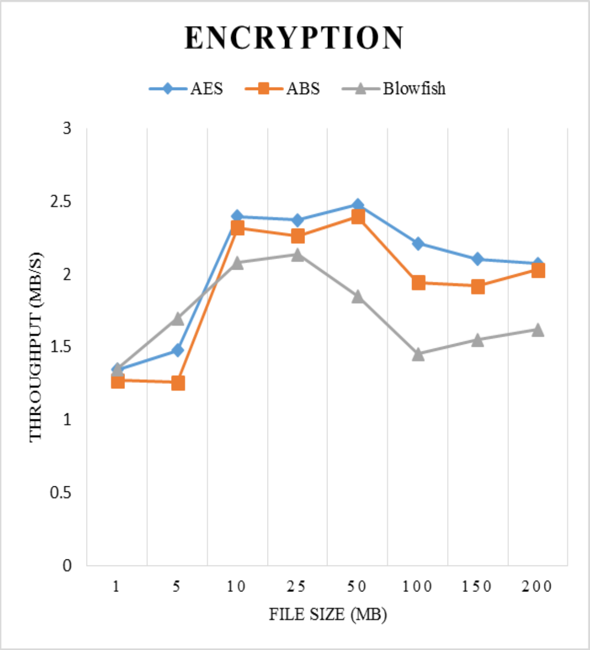

Fig.9. Encryption Throughput Comparison of AES, ABS and Blowfish

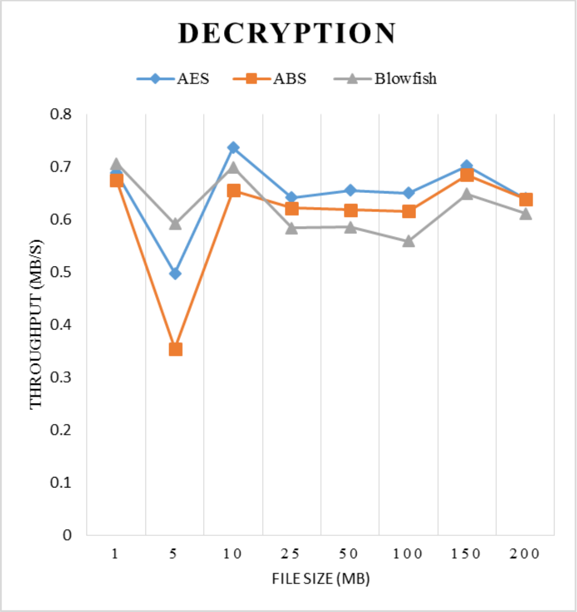

Fig.10. Decryption Throughput Comparison of AES, ABS and Blowfish

In terms of percentage computation overhead of AES with reference to ABS calculated using:

$$\%CO = \left(T_a - T_h\right) \div T_a \times 100\% \, , \qquad (2)$$

where %CO is the percentage computation overhead, T_a is the execution time for AES and T_h is the execution for ABS.

Table 3 below shows the analysis and comparison of the percentage computation overhead of AES to ABS and Blowfish to ABS.

Table 3. Percentage Computation Overhead Comparison of AES and Blowfish with respect to ABS

FILE SIZE (MB)	% Computation overhead of AES – ABS		% Computation overhead of BLOWFISH - ABS	
	Encryption	Decryption	Encryption	Decryption
1	14.67	7.95	23.87	29.31
5	17.25	40.31	34.71	67.00
10	3.26	12.38	-10.36	6.80
25	4.76	3.17	-5.78	-6.14
50	3.41	6.08	-22.77	-5.33
100	13.76	5.69	-25.29	-9.23
150	9.67	2.63	-19.22	-5.21
200	2.01	0.25	-20.30	-4.27

In Fig. 11, the optimum results obtained are approximately 2.01% for the encryption process and 0.25% for the decryption process and the computation overhead tended to zero on increasing the file size. Likewise, the percentage computation overhead of Blowfish was calculated by substituting T_a in (1) with T_b, the execution time of Blowfish. However, the optimum results obtained for Blowfish, in Fig. 12, were about 34.7% during encryption and 67% during decryption but on large file sizes the ABS outperformed Blowfish with a negative percentage overhead of about 25.3% during encryption and 9.22% during decryption.

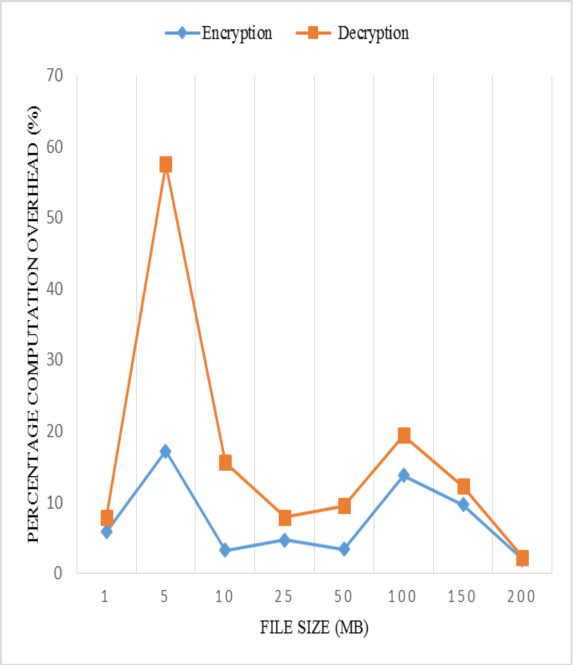

Fig.11. Percentage Computation Overhead of AES Algorithm with Respect to ABS

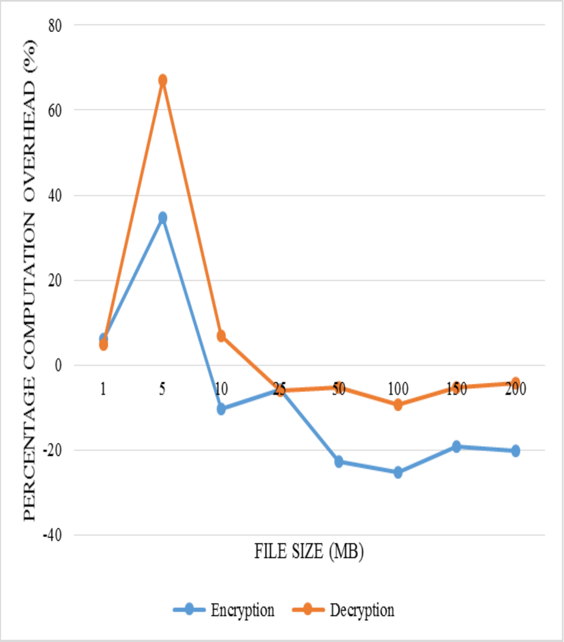

Fig.12. Percentage Computation Overhead of Blowfish Algorithm with Respect to ABS

V. Conclusion

The user data in cloud has been protected with any of symmetry, asymmetry and hybrid cryptographic algorithm to keep it away from an authorized user or theft but the only way to stay confidently secure to aid the full acceptance of MCC is to ensure that users encrypt their data with digital signature before uploading to the cloud. In this paper, a robust and resource-efficient signature based hybrid cryptography of AES, Blake2b and Schnorr signature scheme was proposed as one of the efficient approaches to ensuring confidentiality, integrity, authentication and non-repudiation on resource-poverty devices used in Mobile Cloud Computing.

In the future, the ABS would be compared with other hybrid cryptographic algorithms.

References

[1] N. M. AbdElnapi, F. A .Omara, and N. F. Omran, "A hybrid hashing security algorithm for data storage on cloud Computing", *International Journal of Computer Science and Information Security*, vol. *14, no.* 4, pp. 175, 2016.

[2] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges", *IEEE Communications Surveys & Tutorials*, vol. *16*, no. 1, pp.337-368, 2014.

[3] Y. Alkady, M. I. Habib and R. Y. Rizk, "A new security protocol using hybrid cryptography algorithms", In *Computer Engineering Conference (ICENCO), 2013 9th International*, pp. 109-115, IEEE, December, 2013.

[4] J. P. Aumasson, L. Henzen, W. Meier and R.C.W. Phan, "Sha-3 proposal blake", *Submission to NIST*, 2008.

[5] J.P. Aumasson, W. Meier and R. C. W. Phan, "The hash function family LAKE", In *International Workshop on Fast Software Encryption*, pp. 36-53, Springer Berlin Heidelberg, February 2008.

[6] A. Bhardwaja, G. V. B. Subrahmanyam, V. Avasthi and H. Sastry, "Security algorithms for cloud computing", *Procedia Computer Science*, vol. *85*, pp. 535-542, 2016.

[7] D. De, "Mobile Cloud Computing: Architectures, Algorithms and *Applications", CRC Press, 2016.*

[8] O. Dunkelman and E. Biham, "A framework for iterative hash functions: HAIFA" In *2nd NIST Cryptographic Hash Workshop*, vol. 22, August 2006.

[9] H. M. A Kader, M. M. Hadhoud, S. M. El-Sayed and D. S. AbdElminaam (2014). Performance evaluation of new hybrid encryption algorithms to be used for mobile cloud computing. *International Journal of Technology Enhancements and Emerging Engineering Research*, *2*(4).

[10] V. Kapoor and R. A. Yadav, "Hybrid cryptography technique to support cyber security infrastructure", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET),* vol. 4, no. 11, pp. 3995-4005, November 2015.

[11] A. Kaur, "An age of cloud in Mobile Computing (Mobile Cloud Computing", *International Journal of Advanced Research in Computer and Communication Engineering*, vol. *4*, no. 11, 2015.

[12] A. N. Khan, M. M. Kiah, S. U. Khan and S. A. Madani, "Towards secure mobile cloud computing: A survey", *Future Generation Computer Systems*, vol. *29, no.* 5, pp. 1278-1299, 2013.

[13] P. Kumar and S. B. Rana, "Development of modified AES algorithm for data security", *Optik-International Journal for Light and Electron Optics*, vol. *127, no.* 4, pp. 2341-2345, 2016.

[14] V. Palanisamy and A. Jeneba Mary, “Hybrid cryptography by the implementation of RSA and AES”, *International Journal of Current Research*, vol. *33*, no. 4, pp. 241-244, 2011.

[15] M. N. Rajkumar, J. Muhamed Sabir, V Venkatesa Kumar and S. P. Bharathi, “A comparative analysis of different encryption techniques used in Mobile Cloud Computing”, *International Journal of Research in Advent Technology*, vol. 2, no. 11, November 2014.

[16] P. Patel, R. Patel and N. Patel, “Integrated ECC and Blowfish for Smartphone Security”, *Procedia Computer Science*, vol. *78*, pp. 210-216, 2016.

[17] H. Qi, and A. Gani, “Research on mobile cloud computing: Review, trend and perspectives”, In *Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference on* (pp. 195-202). IEEE, 2012.

[18] Ryan Toukatly. Rochester Institute of Technology SHA-3: The BLAKE Hash Function.

[19] A. Sachdev and M. Bhansali, “Enhancing cloud computing security using AES algorithm”, *International Journal of Computer Applications*, vol. *67, no.* 9, 2013.

[20] C. P. Schnorr, “Efficient identification and signatures for smart cards”, In *Conference on the Theory and Application of Cryptology* (pp. 239-252). Springer New York, August 1989.

[21] W. Stallings, Cryptography and Network Security: Principles and Practice, 6th ed., Pearson Education, 2014.

[22] M. Sujithra, G. Padmavathi and S. Narayanan, “Mobile device data security: A cryptographic approach by outsourcing mobile data to cloud”, *Procedia Computer Science*, vol. *47*, pp. 480-485, 2015.

[23] H. Suo, Z. Liu, J. Wan and K. Zhou, “Security and privacy in mobile cloud computing”, In *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)* (pp. 655-659), IEEE, July 2013.

[24] S. Vanishreeprasad and K N Pushpalatha, “Design and implementation of hybrid cryptosystem using AES and Hash Function. *IOSR Journal of Electronics and Communication Engineering*, *vol.* 10, no. 3, pp. 18-24, doi: 10.9790/2834-10321824.

[25] N. Nagar and U. Suman, “A secure mobile cloud storage environment using encryption algorithm”, *International Journal of Computer Applications*, vol. 140, no. 8, pp. 0975 – 8887, April 2016.

[26] Seyyed Yasser hashemi, Parisa Sheykhi Hesarlo,"Security, Privacy and Trust Challenges in Cloud Computing and Solutions", IJCNIS, vol.6, no.8, pp.34-40, 2014. DOI: 10.5815/ijcnis.2014.08.05

Authors’ Profiles

Oladeji P. Akomolafe, born in 1975, obtained a Master of Science Degree (MSc) in Computer Science at the University of Ibadan, Nigeria in 2004 and a PhD Degree in Computer Science from Ladoke Akintola University of Technology, Ogbomoso, Nigeria in 2014.

His research interests include Pervasive and Mobile Computing, Mobile Agent Technology, Cloud Computing, Context Aware Computing and Software Engineering. He lectures at the Department of Computer Science, University of Ibadan and can be reached at akomspatrick@yahoo.com.

Matthew O. Abodunrin, born in 1988 at Oyo state in Nigeria. He completed his MSc degree in Computer Science at University of Ibadan, Nigeria in 2017 and B.TECH in Computer Engineering from Ladoke Akintola University of Technology, Ogbomoso, Nigeria in 2012.

How to cite this paper: Oladeji P. Akomolafe, Matthew O. Abodunrin,"A Hybrid Cryptographic Model for Data Storage in Mobile Cloud Computing", International Journal of Computer Network and Information Security(IJCNIS), Vol.9, No.6, pp. 53-60, 2017.DOI: 10.5815/ijcnis.2017.06.06